

REMARKS/ARGUMENTS

Applicants' representative would like to thank the Examiner for the interview on December 17, 2008. Proposed amendments were discussed to overcome the rejection under 35 U.S.C. § 103. No agreements were reached.

Claims 1, 11 and 20 are pending in the present application. Claims 2-10, 12-19, and 21-27 were previously canceled; claims 1, 11, and 20 were amended; and no claims were added. Support for the amendments can be found in the claims as originally filed. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 103, Obviousness

The examiner has rejected claims 1, 11 and 20 under 35 U.S.C. § 103 as being unpatentable over Anand (US 2002/0033590 A1) in view of Raventos (US 2002/0194244 A1). This rejection is respectfully traversed. As to claim 1, the Examiner states:

As to claim 1, Anand teaches a method for managing the provisioning of a plurality of resources in a data processing system, said plurality of resources being a plurality of different types (see Abstract, Fig. 2 and 5), said method comprising the steps of:

defining a plurality of provisioning states for each one of said plurality of different types of resources (page 1, [0008], page 2, [0012]-[0013], page 4, [0038], page 5, [0051]);

defining relationships among said plurality of provisioning states, said relationships describing valid transitions from ones of said plurality of provisioning states to other ones of said plurality of provisioning states (page 4, [0040], [0045], page 5, [0059], page 2, [0012]); and defining at least one task that is associated with each one of said valid transitions, wherein defining at least one task that is associated with each one of said valid transitions, comprises (page 2, [0012]-[0013], page 4, [0040], [0045]): specifying a plurality of tasks for each one of said valid transitions (page 2, [0012]-[0013], page 4, [0040], [0045]);

completing said one of said valid transitions for each one of said plurality of different types of resources (standalone computer, notebook computer, hand-held computer, PDA, etc), wherein the same module is used regardless of which resource type is being transitioned (page 3, [0031], [0037], page 5, [0059], page 6, [0064]-[0065]).

Anand does disclose that a workflow comprises of a plurality of processing steps (page 1, [0004], lines 1-14). However, Anand is silent in explicitly teaching the sequence/order of the processing steps or plurality of tasks to complete the transition. Raventos teaches a transaction based service that defines various tasks or functions that could be used on different types of resources, namely, transactional and non-transactional resources such that the sequence order of tasks in a completed transaction is defined/specified as well as the states of the transaction being monitored (see Abstract, page 2, [0009] and [0021], page 6, [0044], page 8, [0055], last 6 lines of [0057], page 10, [0068]). Anand and Raventos are analogous art because they are both in the same field of

endeavor of a transaction/workflow processing system. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Anand such that it would include the feature of providing a sequence/order for completion of the processing steps or plurality of tasks, as taught in Raventos. The suggestion/motivation for doing so would have been to provide the predicted result of aiding to fully and properly activate the services of the system (page 1, last four lines of [0004], [0006], lines 1-5, [0001]).

Therefore, it would have been obvious to one of ordinary skill in the art to combine Anand and Raventos to obtain the invention of claim 1.

Office Action dated September 17, 2008, pages 2-4.

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). The prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In determining obviousness, the scope and content of the prior art are... determined; differences between the prior art and the claims at issue are... ascertained; and the level of ordinary skill in the pertinent art resolved.

Against this background the obviousness or non-obviousness of the subject matter is determined. *Graham v. John Deere Co.*, 383 U.S. 1 (1966). “Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue.” *KSR Int’l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). “*Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.*” *Id.* (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006)).”

Claim 1 is as follows:

1. (Currently Amended) A method for managing the provisioning of a plurality of ~~different types of~~ resources in a data processing system, ~~said plurality of resources being a plurality of different types~~, said method comprising the steps of:
 - defining a plurality of provisioning states for each one of said plurality of different types of resources;
 - defining relationships among said plurality of provisioning states, said relationships describing valid transitions from ones of said plurality of provisioning states to other ones of said plurality of provisioning states;

generating a state diagram for each one of said plurality of different types of resources, each one of said plurality of different types of resources being associated with one of said state diagrams; wherein each one of said state diagrams describing valid transitions for said plurality of provisioning states defined for each one of said plurality of different types of resources; and

defining at least one task that is associated with each one of said valid transitions, wherein defining at least one task that is associated with each one of said valid transitions, comprises:

specifying a plurality of tasks for each one of said valid transitions;

specifying a sequence for completion for said plurality of tasks for each one of said valid transitions, said plurality of tasks being required to be completed in said sequence in order to complete each one of said valid transitions; and

providing said plurality of tasks in said sequence as a module that will complete one of said valid transitions when said module is executed; and

utilizing said module to complete said one of said valid transitions for each one of said plurality of different types of resources, wherein the same module is used regardless of which resource type is being transitioned.

The Examiner failed to state a *prima facie* obviousness rejection against claim 1 because the proposed combination of references, considered as a whole, does not teach or suggest the features of claim 1. Specifically, the proposed combination of references does not teach or suggest the feature of, “generating a state diagram for each one of said plurality of different types of resources, each one of said plurality of different types of resources being associated with one of said state diagrams; wherein each one of said state diagrams describing valid transitions for said plurality of provisioning states defined for each one of said plurality of different types of resources”.

Anand is completely devoid of this feature and teaches the following:

A method for allowing flexible creation and alteration of business processes within a commerce system includes using state machines to describe the actions that can be taken by particular roles at particular points in a process. The state machines are used by a commerce system to enforce validity of user actions, to track the execution of actions within an instance of the business process, to provide the user interface with a list of actions available to a user working on an instance of the business process, to provide coordination between state machines, and to allow different organizations to have varied business processes.

[0008] In typical commerce systems, state information and enforcement of action validity are embedded within the implementation of each of the business processes. Making changes to the business processes is a time consuming endeavor which must be undertaken by system implementers. By modeling and executing business processes as state machines, these processes can be modified without making any changes to the underlying computer programs that are

implementations of the business processes. A commerce function is reconfigured simply by reconfiguring its corresponding state machine. In addition to the functionality of traditional state machines, the present invention adds three key features: the concept of roles, the coordination of interactions of multiple parties, and the ability to allow different organizations to use different versions of the business process.

Anand, Abstract and paragraph [0008].

Anand teaches allowing flexible creation and alteration of business processes within a commerce system using state machines to describe the actions that can be taken by particular roles at particular points in a process. *Anand* teaches modeling and executing business process as state machines. *Anand* is aimed towards a business process. However, *Anand* is completely devoid of teaching or suggesting generating state diagrams. Even assuming, for argument only, that *Anand* does teach utilizing provisioning states, the reference never mentions generating state diagrams describing valid transitions for said plurality of provisioning states.

Additionally, *Raventos* does not teach or suggest this feature of claim 1. *Raventos* teaches the following:

A system and method are disclosed which enable performance of a transaction-based service utilizing non-transactional resources. In one embodiment, a system is provided that includes one or more non-transactional resources. Such resources may, for example, be resources as are commonly implemented within an IDC. The system further includes at least one component that defines one or more tasks executable by at least one of such non-transactional resources. For instance, a plugin component may be developed and communicatively coupled to a resource manager, and such plugin component may define various services (or tasks or "functions") that may be performed on a non-transactional resource. Thus, such resource manager may call the particular functions defined by the plugin component as needed to perform a requested transaction. The system may further include a resource manager operable to control execution of the tasks defined by such component (e.g., plugin component) as a transaction.

Raventos, Abstract.

Raventos teaches using a transaction-based service utilizing non-transactional resources. The system uses components, such as plugins, to define tasks executable by the non-transactional resources. However, *Raventos* is completely devoid of generating state diagrams. *Raventos* does mention "states":

[0057] If the plugin returns successful completion of the task, Resource Manager 405 invokes a CHECK_DO operation of the service to double-check that the operation has actually been performed. Accordingly, at operational block 805,

"xa_prepare" command is executed to cause the tasks stored in the task repository to be checked using the CHECK_DO OP. Thus, the status of the tasks is checked to determine whether they were each successfully completed. If it is determined that all of the tasks were successfully completed, then execution advances to block 806 whereat an "xa_commit" command is executed to make the performance of the tasks by the resource(s) permanent. The XID object for this transaction is then destroyed. If it is determined at block 805 that one or more of the tasks were not successfully completed, then execution advances to block 807 whereat an "xa_rollback" command is executed to rollback the tasks. That is, all tasks are executed with UNDO and CHECK_UNDO OPs to rollback the tasks. Accordingly, if it is determined (e.g., during the CHECK_DO OP) that something went wrong with performance of the task, Resource Manager 405 will execute an UNDO and CHECK_UNDO OPs, to be sure that everything is cleaned up (or undone for the task), and then Resource Manager 405 signals a rollback to the Transaction Manager, which will cause the rest of steps in the transaction to also be rolled back. Thereafter, once the tasks have been committed at block 806 or rolled back at block 807, an "xa_close" command is executed at block 808 to destroy this thread of control, thereby ending performance of the requested transaction. Thus, a transaction may be represented by an XID object, which comprises many different task objects representing the tasks that are to be performed for the transaction. Accordingly, various task objects within a transaction's XID object may each have one of many different states, such as INITIAL, DOING, DONE, CHECKING, CHECKED, UNDOING, UNDONE, or UNDOCHECKING corresponding to the current state of the corresponding task of such transaction.

Raventos, paragraph [0057].

The above portion of *Raventos* teaches that various tasks objects may have different states corresponding to the current state of the corresponding task a transaction. However, the "states" in *Raventos* are not provisioning states. Even assuming, for argument only, that the states of *Raventos* and claim 1 are the same, *Raventos* is still devoid of generating state diagrams.

Because neither *Anond* nor *Raventos* teach or suggest this claimed feature, the proposed combination of references, considered as a whole, does not teach or suggest this claimed feature. Therefore, the Examiner failed to state a *prima facie* obviousness rejection against claim 1. Therefore, the rejection of claims 1, 11 and 20 under 35 U.S.C. § 103 has been overcome.

II. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: December 17, 2008

Respectfully submitted,

/Neil G. Ferrari/

Neil G. Ferrari
Reg. No. 61,484
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants